DOCUMENT RESUME

ED 466 174 IR 021 245

AUTHOR Horiguchi, Tomoya; Hirashima, Tsukasa
TITLE A Framework for Creating Counterexamples in Discovery Learning Environment.
PUB DATE 2001-06-00
NOTE 7p.; In: ED-MEDIA 2001 World Conference on Educational Multimedia, Hypermedia & Telecommunications. Proceedings (13th, Tampere, Finland, June 25-30, 2001); see IR 021 194. Contains small print.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Discovery Learning; *Educational Technology; *Expert Systems; Foreign Countries; Instructional Design; Instructional Development; Instructional Effectiveness; Instructional Materials; Material Development
IDENTIFIERS *Counterexamples; Japan

ABSTRACT

A framework for designing intelligent assistance in a discovery learning environment is proposed in this paper. The process of discovery learning is analyzed and the required functions for intelligent assistance are discussed. A flexible simulator that fits any type of discovery learning is necessary. The problem solvers that perform fundamental tasks of discovery are also needed: hypothesis generator and experiment designer. In particular, the paper focuses on another important function: evaluator of the effectiveness of counterexamples. Counterexamples have many clues for learning, but a learner often feels difficulty in utilizing them. This paper proposes the method of evaluating counterexamples from two educational viewpoints that ask the following: "Does it suggest the occurrence of error clearly?" (visibility), and "Does it suggest the cause of error?" (suggestiveness). Some case studies are presented to illustrate these functions, followed by a description of the framework. (Contains 12 references, 5 figures, and 4 tables.) (Author/AEF)

Reproductions supplied by EDRS are the best that can be made
from the original document.

ED 466 174

A Framework for Creating Counterexamples in Discovery Learning Environment

Tomoya Horiguchi†
†Faculty of Information Engineering
Kobe University of Mercantile Marine
Japan
E-mail: horiguti@ti.kshosen.ac.jp

Tsukasa Hirashima††
††Department of Artificial Intelligence
Kyushu Institute of Technology
Japan
E-mail: tsukasa@ai.kyutech.ac.jp

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY

C.H. Marks

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Abstract: A framework for designing intelligient assistance in discovery learning environment is proposed. The process of discovery learning is analyzed and the required functions for intelligient assistance are discussed. A flexible simulator which fits any types of discovery learning is necessary. The problem solvers which perform fundamental tasks of discovery are also needed: Hypothesis generator and experiment designer. We especially focus on the other important function: Evaluator of the effectiveness of counterexamples. Counterexamples have much clue for learning, but a learner often feels difficulty in utilizing them. We, therefore, propose the method of evaluating counterexamples from two educational viewpoints: (1) Does it suggest the occurrence of error clearly? (Visibility), and (2) Does it suggest the cause of error? (Suggestiveness). Some case studies are presented to illustrate these functions. Then, the whole framework is described.

Keywords: discovery learning, microworld, intelligent assistance, counterexample

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

1. Introduction

Microworld has been getting important as an educational tool to support learning by doing. It provides computer simulation of restricted environment, in which a learner can manipulate the existing objects directly and see the result of her/his action intuitively. A learner explores the world and tries to discover the knowledge and laws in the learning domain. Such a situation is educationally quite good because it promotes a learner's initiative, motivation and interest. Here, discovery learning becomes a central issue.

Discovery learning, however, is not so easy. It needs several skills in 'discovery task,' e.g., how to generate a hypothesis, how to design an experiment to test it. A learner without these skills often comes into an impasse or objectless action, so appropriate assistance is necessary.

One way is to provide some auxiliary tools which makes cognitive process of discovery explicit. For example, in generating hypothesis or in designing experiments, it is quite difficult to find out what are the essential elements of the domain. So, to provide a list of basic variables will be helpful. Hypothesis Editor and Monitoring Tool [Joolingen 1999] are the typical examples.

Another way is to provide more 'intelligent' assistance. It gives a learner some advice concerning the contents of the discovery task, e.g., to suggest a reasonable hypothesis based on the data in hand, to judge the reasonability of the experiment to test the hypothesis. Electric Studio [Shoda 1999] is an example, of which domain is the diagnosis of electric circuit.

For designing the intelligent assistance (which is of our main interest), it needs the problem solvers in discovery task. Especially, hypothesis generator and experiment designer are the essential: The former generates all reasonable hypotheses based on the data in hand, and the latter generates all reasonable experiments to test the hypotheses. They are often depend on the learning domain.

Few discovery learning environments, however, have been developed with such intelligent assistance. It is because there occurs two difficulties in designing such an environment. The one is domain dependency of the problem solvers, but it could be alleviated by modularizing them to interact with other components through abstracted data. SimQuest, the authoring system for discovery learning environment, has the libraries of such components [Joolingen 1997].

The other difficulty is more serious. That is, 'too kind' adviser deprives a learner of her/his initiative, which is the essential merit of discovery learning. (It may teaches her/him what to do next.) Explanations from the adviser must be minimun not to demotivate her/him. In discovery learning environment, it is preferable that the phenomena in microworld themselves make a learner be aware of what to learn.

Such an 'educational' phenomenon often appears as counterexample, which is the phenomenon a learner didn't predict. It impresses on her/him the necessity of learning by suggesting the error in her/his action. Thus, the 'learning from mistakes' is promoted [Perkinson 2000].

A Counterexample, however, must be carefully used in discovery learning. A learner often ignores the anomalous data as the error in measurement, or excludes it out of range of the hypothesis [Chinn 1993]. Even when she/he accepts the counterexample, without any help, she/he cannot reach the correct hypothesis and comes into impasse [Fukuoka 1994, Nakajima 1997].

Therefore, when using counterexamles in microworld, it is necessary to evaluate their educational effective-

IR021245

BEST COPY AVAILABLE

ness and to decide whether they are shown to a learner or not. (Inappropriate counterexamples confuse a learner.) We have studied this kind of 'management' by using Error-Based Simulation (EBS), which simulates a learner's erroneous equation of motion in mechanics [Hirashima 1998, Horiguchi 1998, 1999, 2000]. As a counterexample, EBS must be evaluated from the following viewpoints: (1) Does objects' erroneous motion in EBS make a learner be aware of the occurrence of error? (Visibility), and (2) Does objects' erroneous motion in EBS suggest the cause of error? (Suggestiveness) We have designed such mechanism and developed the EBS management system, of which usefulness has verified through experiments.

In this paper, we propose a framework for designing discovery learning environment with intelligent assistance, especially focusing on the counterexample management. It is done by generalizing our methodology in EBS research project. First, two types of discovery learning is described. Secondly, the required functions for intelligent assistance is discussed. Thirdly, two viewpoints for managing counterexamples are introduced and how to evaluate their effectiveness is also discussed. Some examples are presented for illustration. Lastly, the whole framework is described.

2. Requirements for Intelligent Assistance in Discovery Learning

2.1 Two types of discovery learning

In discovery learning environment, a learner often encounters counterexamples. Figure 1a. illustrates a typical case (type-A). After observing a few phenomena in microworld, a learner constructs a 'theory' which is assumed to rule the world and to explain the phenomena. Then, she/he makes an experiment to test her/his theory, which yields a new instance of phenomenon. If her/his theory is wrong, the new instance contradicts it, to become a trigger of reconsidering the thoery, i.e., counterexample.

In this case, it is assumed that a learner predicts the world of phenomenon ruled by her/his wrong theory, which is compared with the one ruled by 'correct' theory. Most of simulation-based learning environments provide one simulator only which simulates the latter world, because the former already exists in a learner's head.

In some cases, however, the world ruled by a learner's wrong theory needs to be simulated explicitly. For example, when a learner attempts to make an object in microworld move as she/he plans, she/he writes some kind of command sequence, i.e., program. If the program includes some 'bugs,' the object moves in contradiction to her/his prediction. In this case, the world ruled by a learner's wrong theory (eq. program) must be simulated because she/he doesn't predict it, while the correct (planned) world is in her/his head. This is illustrated in Figure 1b (type-B). 'Turtle' world [Papert 1980] is a typical case of this.

Another example of this type is EBS-simulator [Hirashima 1998], which simulates a learner's erroneous equation of motion in mechanics. It is assumed that a learner can correctly predict the motion of objects, but fails to construct correct equation. The world ruled by her/his erroneous equation is simulated and compared with the correctly predicted world.

The difference between these two types of discovery learning mainly comes from the difficulty in formulating a theory. For example, when a learner supposes a proportional relation between two observed variables, it may not be so difficult to formulate it by using linear function. Yet, when she/he predicts an object moves circularly by centripetal force, it will be more difficult to formulate the equation of circular motion. Thus, in the case of complicated formulation, a learner often fails to write her/his theory (eq. prediction) down as the formula. Such a formulation includes several processes and knowledge, so some bugs can easily steal in.

It must be noted that the boundary between these two types often becomes ambiguous. (Suppose that a simple program to control 'Turtle' gradually gets complicated.) It is difficult to know which type of discovery learning is ongoing only from the observation of a learner's action. Therefore, it is necessary to design the flexible simulator which can simulate both correct and erroneous formula of theory. One example is Logo interpreter for 'Turtle' world. EBS-simulator for mechanical world is another example. Both of them allow a learner's formula (Logo program or equation of motion) with some range of semantic (not syntax) error.

2.2 Functions for intelligent assistance

The cause of errors in type-A discovery learning is supposed that a learner dosen't have sufficient data

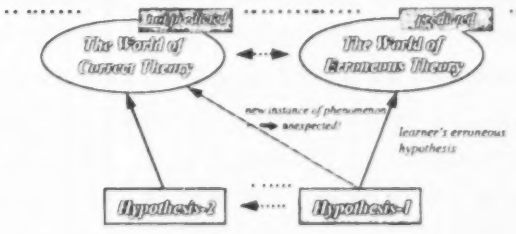


Figure 1a. Discovery Learning (Type-A)

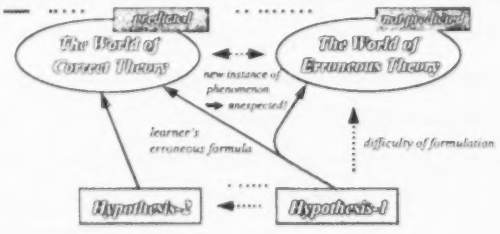


Figure 1b. Discovery Learning (Type-B)

(instance of phenomenon) to make the correct theory, or that she/he mistakes the important data in hand. Therefore, the assistance needed is to suggest the additional experiment which yields counterexample to her/his theory, or to teach how to construct the adequate hypothesis to explain all data in hand.

The cause of errors in type-B discovery learning is, as described above, the difficulty of formulation of theory. The assistance which guides a learner to construct a formula will be helpful.

For providing intelligent assistance in a discovery learning environment, therefore, the following functions are necessary:

Function-1: A problem solver for 'discovery task'

The function which generates all reasonable hypotheses from the given data. It judges the reasonability of a learner's hypothesis based on the data in her/his hand.

It also generates all reasonable actions (i.e. experiments) to verify the given hypothesis, and judges the reasonability of a learner's action based on her/his hypothesis.

In other words, the function as a problem solver for 'discovery task.' (It knows all the 'correct' theory and principle in microworld.)

Function-2: A problem solver for 'formulation task'

The function which generates all formulas of theory in microworld. It knows all the principle which rules the objects' behavior in microworld, and builds them up to the formula.

It also diagnoses a learner's formulation process, and checks the misconceptions within.

In other words, the function as a problem solver for 'formulation task.'

In addition to the functions above, a discovery learning environment needs another important function. A flexible simulator plays an essential role here. When a learner's action/formula yields counterexample, the difference between its result and her/his prediction is very important. It must be clear and meaningful. Even when the action is 'theoretically' reasonable, or even when the formula makes an object's unpredictable motion, if its result isn't clearly different from a learner's prediction, it is not 'practically' (eq. 'educationally') effective. Therefore, the following function is necessary:

Function-3: An evaluator of counterexample's effectiveness

The function which evaluates the effectiveness of the difference between the result of a learner's action/formula and her/his prediction. It is performed as follows: The flexible simulator simulates a learner's wrong hypothesis/formula. It also simulates the correct theory/formula. Then, the both results are compared and the difference is educationally evaluated.

2.3 Two viewpoints to evaluate counterexamples

Apparently, it is necessary to prepare some kind of criteria with which the difference of two results is evaluated. 'Visibility' is the key issue in considering this. Because all a learner can observe in microworld is the objects' behaviors (i.e. phenomena), her/his erroneous idea must be visualized in them. There are two viewpoints:

Viewpoint-1: Awareness of the occurrence of error

In this viewpoint, it is made sure that two simulated results are clearly different. It is measured by comparing the specific (often physical) variables of these phenomena in microworld by using the criteria, which defines what kind of difference of the variables is 'visible' to a learner.

This suggests a learner's hypothesis/formula contains some erroneous idea, to motivate her/him for reconsidering.

Viewpoint-2: Awareness of the cause of error

In this viewpoint, it is made sure that the difference of two simulated results points out the cause of error, besides its occurrence. It suggests a learner how to correct her/his erroneous idea. The criteria is necessary, which defines what kind of difference of the variable in microworld suggests what kind of error in problem-solving.

The former viewpoint is concerned only with the phenomena themselves in microworld. Some observable (often physical) variables are selected and checked whether they have clearly visible difference between the two results. The ability of human perception should be carefully considered to define the 'visibility.' (It may be useful to provide some kind of 'visual tool' which aids a learner to observe the variables.)

This viewpoint is comparatively simple, but often useful to give a learner good motivation for reconsidering.

The latter viewpoint is more complicated. It is concerned not only with the phenomena themselves but also with the problem-solving process. The difference between the two results must be 'suggestive' of the cause of a learner's error. (Here, to be 'visible' is a necessary condition.) Some kind of rules are necessary, which link the phenomena in microworld to the misconceptions in problem-solving. The criteria should be defined based on the task analysis of the domain.

The phenomena which are 'visible' but not 'suggestive,' sometimes mislead a learner because it doesn't reflect her/his problem-solving process. This viewpoint becomes necessary in such cases.

We call both 'visibility' and 'suggestiveness' in above two viewpoints 'visibility' sensu lato. In defining the

criteria for Function-3, 'visibility' plays an important role. It is much concerned with the problem-solving process of Function-1 and Function-2, so these three functions (and the flexible simulator) must cooperate with each other.

In the next chapter, we present some examples which illustrate how the criteria are defined and how the effectiveness of counterexamples are evaluated. Then, we propose a general framework of designing the discovery learning environment with intelligent assistance discussed above.

3. Examples: designing the evaluator of counterexamples' effectiveness

In this chapter, we describe how to provide intelligent assistance when a learner encounters counterexamples. It is illustrated by using a series of case studies in managing Error-Based Simulation (EBS) [Hirashima 1997, Horiguchi 1999, Horiguchi 2000]. EBS simulates a learner's erroneous equation of motion in mechanics. Objects' unnatural motion in EBS often differs from their correct motion predicted by a learner, which motivates her/him for reconsidering her/his erroneous equation.

The difference, however, is not usually 'visible' to a learner, or sometimes misleads her/him. Therefore, it is necessary to evaluate the effectiveness of EBS as a counterexample. We've designed such mechanism and developed the EBS management system. Thus, what we discuss here is type-B discovery learning, and is mainly concerned with Function-3 described in chapter 2.

3.1 EBS management from viewpoint-1 [Hirashima 1998]

In the earlier stages of our research, EBS was mainly managed from the viewpoint-1. In motion simulation of physical objects, the most specific variable is their velocity and acceleration. So, using these variables, we defined the following criteria for evaluation of the difference.

Criterion for Error-Visualization-1 (CEV-1):
In EBS, an object's velocity must *qualitatively* differ from its correct velocity, i.e., their qualitative values must be different.

Criterion for Error-Visualization-2 (CEV-2):
In EBS, the rate of change of an object's velocity must *qualitatively* differ from the one of its correct velocity, i.e., their qualitative values must be different. The rate of change of velocity usually means acceleration, but sometimes means the derivative by non-time parameter.

These criteria are derived from the fact that human ability of vision is sensitive to rather the qualitative properties of motion than the quantitative ones. CEV-1 is preferred to CEV-2 because the velocity is easier to perceive than its derivative. The reasonability of them were verified through cognitive experiments [Horiguchi 1998].

By using these criteria, the effectiveness of EBS is evaluated as follows: When an erroneous equation is constructed by a learner, it is simulated by EBS-simulator. The correct equation is also simulated, and it is checked whether the difference between the two simulations satisfy CEV-1 and/or CEV-2. The module which performs this process is implemented by using qualitative reasoning techniques, and called EBS-manager [Hirashima 1998].

For example, consider the problem in Figure 2a. When a learner constructs Equation-B, the EBS which simulates the equation is judged 'effective,' because the block's velocity and acceleration in the EBS are qualitatively different from the ones in correct simulation. (It satisfies both CEV-1 and CEV-2.)

When a learner constructs Equation-C, matters are more complicated. The EBS which simulates the equation satisfies neither of CEVs by itself, because the block's velocity and acceleration in it are *not qualitatively but only quantitatively* different from the ones in correct simulation. EBS-manager, in such a case, searches some 'modification' of the EBS which makes qualitative difference. In this case, perturbing the angle of slope Ø works well. When Ø increases, the block's velocity decreases according to Equation-C, while it increases according to the correct equation. This satisfies CEV-2, so such an EBS is generated and used as a counterexample. The snapshots of the EBS management system are shown in Figure 2b.

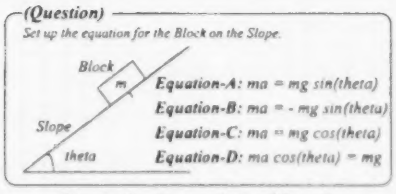


Figure 2a. Example Problem-1

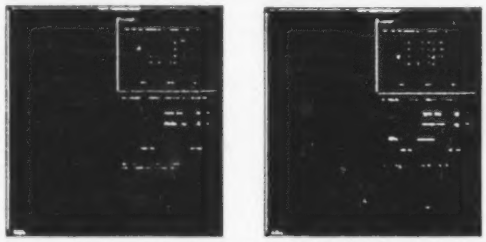


Figure 2b. Error-Based Simulation

Table 1. Force-Enumerating Rules (FERs) (abstract)

force	Rules for enumerating forces		
friction	R3: r3-c1	Object-1 and Object-2 are touching together	• •
	r3-c2	coefficient of friction of touching surface $\mu > 0$	• •
	r3-c3	normal force N acting on touching surface	• •
	r3-c4	Object-1 and Object-2 are moving oppositely along the tangent	
	• •		
	r3-a1	friction Ff1 to Object-1	**Qualitative**
	r3-a2	friction Ff2 to Object-2	**Qualitative**
	r3-a3	Direction(Ff1): opposite to the velocity of Object-1	**Qualitative**
	r3-a4	Direction(Ff2): opposite to the velocity of Object-2	**Qualitative**
	r3-a5	Magnitude: Ff1 = Ff2 = μN	**Quantitative**

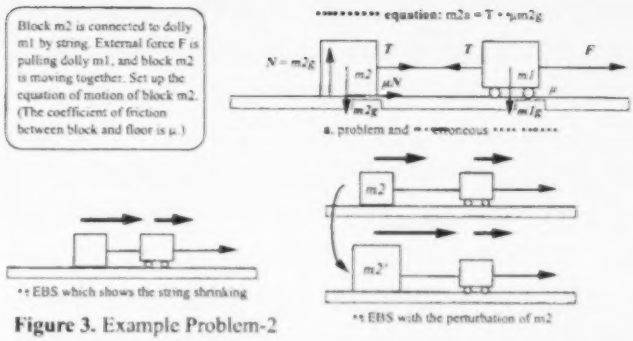

Block m2 is connected to dolly m1 by string. External force F is pulling dolly m1, and block m2 is moving together. Set up the equation of motion of block m2. (The coefficient of friction between block and floor is μ.)
equation: m2a = T • μm2g
N = m2g
T
T
F
m2
μN
m1
μ
m2g
m1g
a. problem and • erroneous
m2
m2'
•• EBS which shows the string shrinking
•• EBS with the perturbation of m2

Figure 3. Example Problem-2

Table 2. Error-Identification Rules: EIRs (abstract)

force	appearance	cause of errors	correcting strategy
friction	missing	missing knowledge of friction (**R3**)	re-teach the concept/definition
		overlooking the touching together (**r3-c1**)	re-show the problem and indicate the corresponding part
		overlooking that coefficient of friction $\mu > 0$ (**r3-c2**)	re-show the problem and indicate the corresponding part
		missing normal force (**r3-c3**)	proceed to the correcting strategy of normal force
		belief that normal force doesn't work (**r3-c4**)	indicate that friction is missing
	extra	missing that coefficient of friction $\mu = 0$ (**r3-c2**)	re-show the problem and indicate the corresponding part
		extra of normal force (**r3-c3**)	proceed to the correcting strategy of normal force
		belief that normal force works (**r3-c4**)	indicate that friction is extra
		extra of the force which causes friction (**r3-c4**)	proceed to the correcting strategy of that force
	error	error of normal force (**r3-c3**)	proceed to the correcting strategy of normal force
		error of the force which causes friction (**r3-c4**)	proceed to the correcting strategy of that force
		error of direction/magnitude (**r3-a3/4/5**)	indicate that direction/magnitude is erroneous

Table 3. Criteria for Cause-of-Error Visualization: CCEVs for single object (abstract)

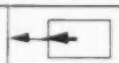
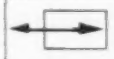
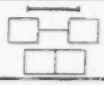
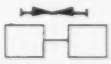
motion		difference	suggesting errors
correct motion		•	• • •
motion (c)		velocity: same acceleration: opposite	missing of the force same as moving direction • • extra of the force opposite to moving direction • • smaller of the force same as moving direction • • larger of the force opposite to moving direction • •

Table 4. Criteria for Cause-of-Error Visualization: CCEVs for two objects (abstract)

motion		unnaturalness	suggesting errors
correct motion		constant distance	• •
motion (e)		closing string shrinks	extra/larger of the tension • • extra/larger of the propagating force • •

3.2 EBS management from viewpoint-2 [Horiguchi 2000]

The merit of viewpoint-1 is its simplicity. It does not depend on the problem-solving process but only on the resulting phenomena, so it is comparatively easy to design the evaluator of counterexamples' effectiveness.

The counterexamples evaluated from viewpoint-1, however, don't always provide a learner useful information to correct her/his error, and sometimes mislead her/him. This comes from the lack of consideration of the problem-solving process. Therefore, our recent researches are paying more attention to the viewpoint-2 for managing EBS.

Apparently, the problem solver which can construct the correct equation is necessary. (It means the Function-2 in chapter 2.) We developed it by modelling the formulation process of equation in mechanics. The model

focuses mainly on the process in which a learner enumerates the forces acting on the objects, so it consists of a set of production rules, called Force-Enumerating Rules (FERs). They describe the conditions for the forces to act. Part of them are shown in Table 1.

The correct solution inferred by the problem solver is compared with the one by a learner (inputted through the interface which allows her/him to construct equations and diagrams), and the differences are checked. Some rules are necessary which link the error-appearance on her/his solution to its cause. By considering a learner's error as the error about FERs, the rules are formulated as shown in Table 2. They are called Error-Identification Rules (EIRs), which link the erroneous part of a learner's solution to its cause and instruction strategy.

The identified cause of error must be visualized and suggested by EBS. The last set of rules we need is the one which describes what kind of motion in EBS suggests what kind of misconception in problem solving. The fundamental idea is quite simple. When a human observes an object moving, she/he feels its 'motive force' working. We apply this fact to the difference of motion. For example, assume that a learner observes a block moving to the left with deceleration when she/he predicted it moves to the left with acceleration. She/he will feel that the force which acts to the left is missing, or that the force which acts to the right is extra. The same thinking is possible about the relative motion of two objects. The rules are formulated as shown in Table 3 and 4. They are called Criteria for Cause-of-Error Visualization (CCEVs), which link the motion in EBS to the cause of error suggested.

For example, consider the problem in Figure 3. When a learner constructs the equation in Figure 3a (the direction of friction μN is erroneous), the 'unmodified' EBS becomes as shown in Figure 3b, in which the string shrinks. According to the CCEV in Table 4, however, this motion suggests the error about tension, which is not the cause of error in this case. EBS-manager, therefore, searches the 'modification' which makes EBS suggest the real cause of error. It finds the perturbation of the mass of block m2 works well. When the mass m2 increases, the block's velocity increases according to the erroneous equation, while it decreases according to the correct equation. This satisfies the CCEV in Table 3, and such an EBS correctly suggests the error about friction μN (Figure 3c).

4. Framework for Discovery Learning Environment and Concluding Remarks

The functions which are required for intelligent assistance in discovery learning environment were discussed. They are: the problem solver for discovery task, the problem solver for formulation task, the flexible simulator, and the evaluator of the simulations' effectiveness. (The last needs some criteria for evaluation.)

The more 'intelligently' each function is designed, the more highly it depends on the learning domain. So, it is preferred modularizing them as independent components to designing them according to some common template. This modular model allows the system designer to utilize the existing tools and simulators of each domain. The intelligent adviser will adjust these modules and user interface (which should be also modularized). The one key issue is to abstract the variables each module uses for efficient interaction, and the other is to enrich the component libraries.

We are now developing the authoring system for intelligent discovery learning environment of such an architecture.

References

[Chinn 1993] Chinn, C.A. & Brewer, W.F.: Factors that Influence How People Respond to Anomalous Data, Proc. of 15th Ann. Conf. of the Cognitive Science Society, pp.318-323.

[Fukuoka 1994] Fukuoka, T. & Suzuki, K.: A Case Study on Lower Secondary School Pupils' Conceptual Changes through Demonstration of Counterexamples: The Concept of Dynamics in the Case of Roller Coaster Models, Bulletin of Society of Japan Science Teaching, Vol.35, No.2, pp.21-32 (in Japanese).

[Hirashima 1998] Hirashima, T., Horiguchi, T., Kashihara, A. & Toyoda, J.: Error-Visualiza-tion by Error-Based Simulation and Its Management, Int. J. of Artificial Intelligence in Education, Vol.9, No.1-2, pp.17-31.

[Horiguchi 1998] Hirashima, T., Horiguchi, T., Kashihara, A. & Toyoda, J.: Two Points of View for Managing Error-Based Simulation, Proc. of ICCE98, pp.422-431.

[Horiguchi 1999] Hirashima, T., Horiguchi, T., Kashihara, A. & Toyoda, J.: Error-Visualization by Error-Based Simulation Considering Its Effectiveness -Introducing Two Viewpoints, Proc. of AI-ED99, pp.421-428.

[Horiguchi 2000] Horiguchi, T. & Hirashima, T.: A Method of Creating Counterexamples by Using Error-Based Simulation, Proc. of ICCE2000 (to be appeared).

[Joolingen 1997] van Joolingen, W., King, S. & de Jong, T.: The SimQuest Authoring System for Simulation-Based Discovery Learning, Proc. of AIED97, pp.79-86.

[Joolingen 1999] van Joolingen, W.: Cognitive Tools for Discovery Learning, Int. J. of Artificial Intelligence in Education, Vol.10, No.3-4, pp.385-397.

[Nakajima 1997] Nakajima, N.: The Role of Counterevidence in Rule Revision: The effects of instructing metaknowledge concerning non adhocness of theory, Japanese Jour. of Educational Psychology, 45, pp.263-273 (in Japanese).

[Papert 1980] Papert, S.: Mindstorms -Children, Computers, and Powerfull Ideas, Basic Books, Inc., New York.

[Perkinson 1984] Perkinson, H.J.: Learning from Our Mistakes -A Reinterpretation of Twentieth-Century Educational Theory, Greenwood Publishing Group.

[Shoda 1999] Shoda, K., Kusano, R., Nakamura, M. & Otsuki, S.: Assisting Discovery Learning in Electric Studio, Proc. of ICCE99, pp.867-874.

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

NOTICE

Reproduction Basis

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").